RHQ GRAPHING

with d3.js

D3 HISTORY

- Mike Bostock designed d3 to be a replacement for protovis: http://mbostock.github.com/protovis/ex/.

- Similar capabilities just totally different implementation

WHAT DOES D3 STAND FOR?

- Data
- Driven
- Documents

WHAT IS D3?

- d3 is a framework/toolkit for DOM manipulation
- Its like: jQuery for visualizations (more than charts)
- It is low level -- not an API -- but powerful
- Each chart type can have its own js library and syntax and data model (changing a chart type means all of this can change)
- Uses existing standard technologies such as CSS3, html5, svg and canvas
- d3 binds data (from json or csv) to html, canvas, webGL or svg

WHAT CAN I DO WITH D3?

- Do I have to know html, svg or canvas -- Yes, its that low-level. Did I mention its not an API?!
- Can I do animations? Yes. It has a very rich syntax for dealing for binding data to a) new objects (enter), b) existing objects (update) and c) removing (delete)old data/objects from a visualization
- Essentially, its a data mapper DSL for binding json/csv to svg/canvas/html
- It takes care of things like making sure my graph fits proportionately within my bounding viewport (think scaling)

WHY DID WE CHOOSE IT?

- Because its COOL!
- Don't believe me? Try it: https://github.com/mbostock/d3/wiki/Gallery
- The visualizations are amazing. It is much more than a charting library. It gives us something we can grow into over time. Our charts will only get better.

D3 SPECIFICS

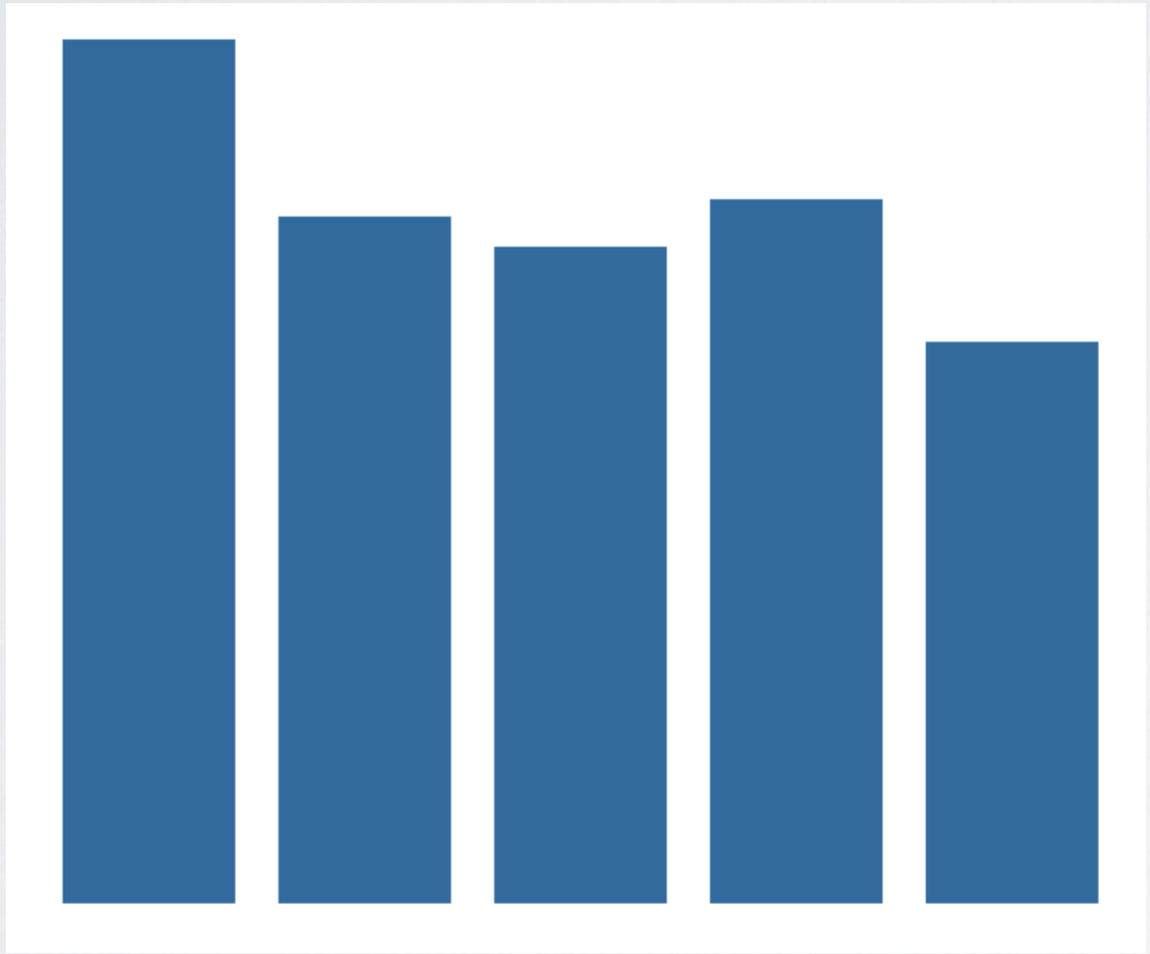
- loading and parsing: json, csv, tsv(tab separated)

```
d3.csv("sp500.csv", function(data) {


d3.select("#example")
    .datum(data)
    .call(chart);
});
```

- binding data to html dom
- colorscales - professional coordinated color schemes
- scaling value domains to pixel area

BAR CHART EXAMPLE

http://www.recursion.org/d3-for-mere-mortals/

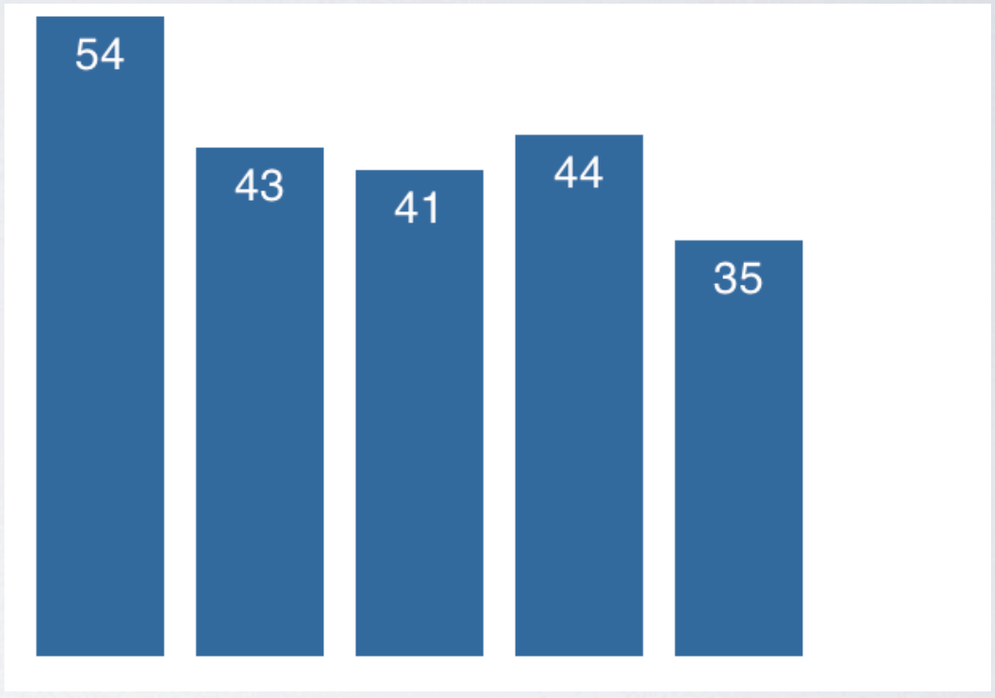


CODE WALK-THROUGH

```
var data = [{year: 2006, books: 54},
            {year: 2007, books: 43},
            {year: 2008, books: 41},
            {year: 2009, books: 44},
            {year: 2010, books: 35}];

var barWidth = 40;
var width = (barWidth + 10) * data.length;
var height = 200;

var x = d3.scale.linear().domain([0, data.length]).range([0, width]);
var y = d3.scale.linear().domain([0, d3.max(data, function(datum) { return datum.books; })]).
  rangeRound([0, height]);

// add the canvas to the DOM
var barDemo = d3.select("#bar-demo").
  append("svg:svg").
  attr("width", width).
  attr("height", height);

barDemo.selectAll("rect").
  data(data).
  enter().
  append("svg:rect").
  attr("x", function(datum, index) { return x(index); }).
  attr("y", function(datum) { return height - y(datum.books); }).
  attr("height", function(datum) { return y(datum.books); }).
  attr("width", barWidth).
  attr("fill", "#2d578b");
```

ADD TEXT LABELS

```
barDemo.selectAll("text").
  data(data).
  enter().
  append("svg:text").
  attr("x", function(datum, index) { return x(index) + barWidth; }).
  attr("y", function(datum) { return height - y(datum.books); }).
  attr("dx", -barWidth/2).
  attr("dy", "1.2em").
  attr("text-anchor", "middle").
  text(function(datum) { return datum.books;}).
  attr("fill", "white");
```



LIFE CYCLE

- Enter: Add new elements to the stage
- Update (default): do something to existing elements on the stage
- Exit: remove the existing elements from the stage (perhaps via a transition). Opposite of enter. Contains all the elements with no corresponding data
- Example: http://www.jeromecukier.net/projects/agot/places.html

HANDS-ON

- http://enjalot.com/tributary/3941260/

D3 CREATION STEPS

1) make sure data is in proper format (json, csv)	`var data = [{year: 2006, books: 54},`
	`{year: 2007, books: 43},`
	`{year: 2008, books: 41},`
	`{year: 2009, books: 44},`
	`{year: 2010, books: 35}];`

2a) select or selectAll
2b) bind data to dom
2c) append elements
2d) fill in attributes and style
	`#NOTE: this example has the data self contained`
	`# for demo purposes`
	`svg.selectAll("circle")`
	`.data([32, 57, 112, 293])`
	`.enter().append("circle")`
	`.attr("cy", 90)`
	`.attr("cx", String)`
	`.attr("r", Math.sqrt);`

3) set x and y scales and axes (map domain into range)	`var x = d3.scale.linear().domain([0,`
	`data.length]).range([0, width]);`
	
	`var y = d3.scale.linear().domain([0,`
	`d3.max(data, function(datum) { return`
	`datum.books; })]).rangeRound([0, height]);`

INTERACTIVE EXAMPLES

- http://www.jeromecukier.net/ - sophisticated interaction

SO WHY ISN'T EVERYONE USING D3?

- Must think at the low level. Coordinate calculations are required
- Different chart types can be totally different to implement (different authors, philosophies, etc...). Remember we are building charts from a toolkit not an API
- More difficult to use than an API graphing solution; because it is a framework/toolkit
- No GWT wrapper (for the GWT inclined)
- Oh yeah, and No IE8 (and prior) support!

IS NVD3 FOR YOU?

- nvd3.js is a higher level API that sits atop d3
- It provides a reusable experience by trying to standardize the data model and chart types (thereby lowering the learning curve)
- Each graph type doesn't require totally different setup
- No need to do geometry calculations and learn SVG
- This is more the API experience that existing charting APIs support (instead of the DIY experience that d3 brings)
- Live Examples: http://nvd3.com/livecode/

GOTCHAS

- Non-unique tag ids in a page ( ). A problem for multiple graphs or singe page apps.

- In Json, dont store values in the keys

- IE8 compatibility -- This should be addressed first before you pick graph type and discover the tag is not supported by your compatibility solution. Lots of solutions covering 85% scenario don’t read about a solution and assume it will work with your graph type. TEST each chart type!